micro-layouts

Surfaces, Shape & Micro Layouts Every component is not just present Poised, Purposeful, & Human If macro layouts are the skeleton of our experience, then surfaces, shapes, and micro layouts are the tangible flesh and muscles users engage with. This guide defines how we construct component structures—both their physical contours and internal arrangements—ensuring our Intuitive and Engaging pillars are consistently reflected. This is where two of our signature souls— Glass and Fluid —converge: Surfaces Surfaces create a modern, layered interface through elevation and depth. Shapes Shapes express tactility, tone, and interaction cues via soft, fluid geometries. Micro Layouts Micro layouts arrange content inside components with deliberate rhythm and hierarchy. Core Principles ■ Principle Design Guideline Developer Implementation Harmonious Maintain consistent spacing and corner radii using tokens. Use `$space-*` and `$shape-radius-*` tokens. Avoid manual pixel values. Contained Components must be self-contained and modular. Apply internal padding and avoid relying on parent context for structure. Intentional Eliminate unnecessary decorative elements. Audit frequently; ensure surfaces, corners, and layouts serve UX purposes. Scannable Structure content for effortless reading and visual flow. Use F-patterns, Z-patterns, and `flex/grid order` properties to guide attention. Inclusive Do not rely solely on shape or depth to communicate state or interaction. Pair shapes with color, icons, text. Test contrast, particularly on glass surfaces. The Surface System: Depth & Glass ■ Surface Levels ■ Level Description Typical Usage Surface-0 Base, opaque background. Application background Surface-1 Slightly raised, common interaction layer. Cards, buttons, input fields Surface-2 Floating, elevated components. Modals, dialogs, dropdowns Glassmorphism ■ Glass effects introduce subtle blur and transparency for a modern aesthetic. Apply only to Surface-2 or elevated Surface-1 elements like sidebars Transparency : Use $color-background-glass token. 5% Transparency 10% Transparency 20% Transparency Backdrop blur Use $color-background-glass token. Edge definition : Apply a 1px semi-transparent border. No Border 1px Semi-transparent Shadow Tokens ■ Token Value Use Case `$elevation-1` `0px 2px 4px rgba(0,0,0,0.08)` Cards, inputs `$elevation-2` `0px 4px 12px rgba(0,0,0,0.10)` Modals, dropdowns `$elevation-glass` `0 4px 24px rgba(255,255,255,0.06)` Under glassmorphic containers The Shape System: Contour & Feel ■ Shape Hierarchy ■ The more interactive or emotionally significant a component is, the more distinct its shape: Subtle rounding For structure and content organization Examples: Card Example Input Field Cards, inputs, content containers Full rounding For direct interaction and emphasis Examples: Tag JD Chips, avatars, action buttons Border Radius Tokens ■ Token Value Usage Examples `$shape-radius-sm` 8px Buttons, modals, alerts, tooltips `$shape-radius-md` 12px Cards, inputs `$shape-radius-lg` 24px Expressive containers, emphasis cards `$shape-radius-pill` 9999px Chips, tags, icon buttons `$shape-radius-circle` 50% Avatars, circular indicators Common Component Patterns ■ Card Surface: Surface-1 Shape: $shape-radius-md or $shape-radius-lg Padding: $space-sm to $space-md Button Surface: Surface-1 Shape: $shape-radius-sm or $shape-radius-pill Padding: Based on size variant, uses spacing tokens Modal Surface: Surface-2 with Glass effect Shape: $shape-radius-sm or $shape-radius-md Padding: $space-lg (32px) Micro Layout Patterns ■ Micro layouts define how internal content is organized within components. These patterns enhance rhythm, scannability, and composability: Stacked Group Use Case: Forms, setting panels Layout: Vertical stack with equal spacing, optional separators Tokens: $space-md between items Tile Grid Use Case: Dashboards, overview cards Layout: Responsive grid (2–4 cols) with consistent gaps Tokens: Grid gap via $space-lg Media + Text Block Use Case: Profile summaries, feature previews Layout: Horizontal row with image and content stack. Responsiveness: Stack vertically below md breakpoint. Tokens: Stack vertically below md breakpoint Section Container Use Case: Grouped forms, onboarding screens Layout: Light background (Surface-1 or themed container) Tokens: $space-lg inside, $shape-radius-md on corners Inline Controls Row Use Case: Filter bars, table actions Layout: Horizontal with end-alignment Tokens: Left/right padding via $space-sm, center aligned vertically Mobile & Tablet Consideration ■ Shape consistency Token-based radius remains the same across devices. Surface economy Prefer flatter designs or full-width components on small screens.

Modal behavior Use full-screen modal variants for mobile. Performance fallback On low-power devices, disable backdrop filters. Guardrails ■ Do Use tokens for all spacing, elevation, and shape. Don't Eyeball spacing or corner radii. Do Apply Glass effect only where needed. Don't Use glassmorphism everywhere. Do Maintain accessible contrast on frosted surfaces. Don't Assume blur is sufficient for legibility. Do Match shape size to interaction significance. Don't Use shape alone to imply interactivity. Do Use container spacing inside components. Don't Depend on parent padding to define component boundaries. Shape, surface, and internal layout are more than visual decoration—they are the rhythm, breath, and soul of Play+. Applied consistently, they elevate UI into experience.